

# DCC++ EX Project Meeting Minutes

Tuesday July 7, 2020 11:30AM EST Ending 12:48

## Attendance

Larry Dribin  
Roger Beschizza  
Harald Barth  
Fred Decker  
Anthony Williams (Dex++)  
Kevin C. Smith  
Gregor Baues  
Steve Todd  
and other visitors

After a quick thank you to all the team members who have done so much great work to this point the following topics listed old business were discussed

## Feasibility of the library approach to developing

The majority still feel that the development of key portions of the code such as the waveform generator and packet generator should be a library. People are used to having to install libraries in the Arduino IDE and the installer software makes it a moot point. The library will be easier to maintain and simplify the code for Users.

Anthony (Dex++) gave a demonstration of the compiler handling building the new Command Station EX version and then switching to Visual Studio to demonstrate how he has the installer able to handle everything.

*Status: Implemented*

## Coding Standards

In order to maintain uniformity and readability of the code, we setting on the Google method. The CONTRIBUTING.md file in each repository explains this to those wishing to help with coding on this project. Those trying to issue a pull request will see the link to the contributing guidelines in a sidebar to the right.

*Status: Complete. May need review once a quarter.*

## Removing Code Comments

We will take care of this in the next release. All of the extra comments are being put into a DCC++ EX Manual and on the website.

*Status: In process*

## Reducing Thread Size on TrainBoard

By moving our technical conversations to Discord, we were able to slow down TrainBoard traffic and make the threads more readable for new users.

*Status: Complete*

## Railcom Support

The Railcom cutout is working and people with Railcom equipment can communicate with their devices while still running trains with DCC++ EX. This may be enough, but we will revisit at some point to see if reading Railcom data in the Command Station is something we want to do. Chris mentioned that he felt Railcom was used mostly for block detection and handling this presented complications.

*Status: Ready for testing*

## Networking / Engine Driver

These items are grouped together because one of our priority items is being able to create a WiThrottle server in the Command Station and be able to use Engine Driver to run trains without any other hardware or software.

There are two primary candidates for a simple enough solution to support our “Operators”, a Mega WiFi or an ESP8266 shield made by Makerfabs and sold by Robotshop. The issue with the WiFi Mega is that the firmware on the ESP8266 can be different between the 3 boards. We need to test which brands definitely have the standard and working AT firmware so that users don’t have to flash it. That said, Fred (FflightRisk) found firmware and slugged through the process of using the Expressif Flash Tool to be able to get it working. So we can provide instructions for those who need to flash an ESP8266. There is also a python application that can do this, so we can script this or perhaps integrate it, or the equivalent, into our installer

Chris (UKBloke) is using the WangTongze Wifi Shield though pins have to be bent or you can jumper to it off of the Arduino

Fred used an unbranded Mega WiFi which would not work out of the box. He found out how to flash these chips which requires multiple files and setting entry point addresses. Not for the faint of heart, but relatively easy if you give someone the files and a picture of the settings they need to match.

Fred also tried the tiny ESP-01S boards and they work with no changes, but you have to solder or jumper four wires.

Steve Todd is working with us to be able to provide a robust implementation. He has other ideas for what we can add to this, like rosters. We will cover more in a later meeting.

*Action: Compile list of Steve’s suggestions into a todo list for the Throttle project. Get more people testing.*

NOTE: We are investigating the possibility of creating our own Wifi board.

## Method for handling turnouts, accessories

We only touched the surface of the methods we can use for control. Chris wrote that he uses an I2C bus and that the issue is with JMRI. Their tag language only deals with DCC addresses for turnouts. We also

need multiple turnout types on the same layout, so can't just use a global setting. TPL has a way to address turnouts, but this may conflict with JMRI.

### External API

We had a verbal presentation from Gregor (Grbba) about his progress with using the external API we can call into that uses MQTT, a part of "The Internet of Things" (IOT) and one of the ways smart home devices communicate to each other. He has his test case working. Using the cloud or a local network, this system could control anything from anywhere by sending simple messages. Gregor will follow up with a paper on basic operation and where we can take this.

*Action: Gregor will give us a white paper and sample code for us to test in the next 2 weeks*

### JMRI Integration

Some of our new features will require changes to JMRI. Fred, Anthony (Dex++) and Harold have looked at the JMRI code. Steve Todd can help with contacts that are more familiar with the Java code (including Steve himself). We have more information to try one more time to see if we can contact Twindad.

Chris wrote that we have to deal with having multiple communication protocols. There is a board posted issue about JMRI not handling the "<p2>" and "<p3>" messages properly. Our track power monitors are not aware of JMRI and we certainly won't always be using JMRI as we could be using things like WiThrottle/Engine Driver or TPL. We need to look at USB communication vs. WiFi or Ethernet.

*Action: Create a list of changes/additions we need to implement. Follow up about note mentioning throttle with sockets or REST*

### Modbus Support

*Status: tabled*

### Software Switching of Main and Program tracks

Chris felt this was something we should do and Anthony seemed pretty passionate about it. Larry made a great case as well. Harald mentioned concerns about the DCC waveform being out of phase during the switch on the older DCC++ systems, but that we may be able to do it on the new platform.

*Action: Investigate what it will take and see if we can try some test code*

### Current Sense / ACK Detection

Fred and Harald had independently and together worked on different parts of the problem. Harald created a method for checking in between packet sets. There has been a great collaboration on Discord working on all the details and Chris and Harald are making progress. There are many variables with different motorboards and voltages. This is the number one technical support issue with the old DCC++

*Status: We have working ACK detection and better current sense.*

*Action: Continue testing actual cases with resistors on the track.*

*Assigned: Chris, Harald, and Fred*

## New Business

### Installer

*Action: Start incorporating more message to let users know what is happening.*

*Assigned: Anthony*

### Website

Fred mentioned how the “path system” will work to provide different presentations to Operators, Tinkerers and Engineers.

*Action: Add more content and create a flow diagram for the website with clickable links to be able to jump to any path*

*Assigned: Mani, Fred*

### Licensing

Had a brief discussion of two kinds of licensing we might need to cover code or libraries imported into DCC++ EX and those we will provide for open source

*Action: Come up with the final licensing*

*Assigned: Chris, David, Harald, Fred*

### TPL (Train Programming Language)

*We spoke briefly about TPL and how it will work*

### FireBox

New boards will be here sometime in July. We will get one to Steve Todd for testing and work on how to put them together and sell them.

### Top Priorities

- Get networking issues resolved
- Test and advance Engine Driver
- Move ahead with FireBox (get one to Steve Todd)
- JMRI Integration

*Action: Get 'er done!*